初步探索评估系统

2022-09-15T15:06:13+08:00

探索一下这个系统的功能都有哪些，它们在代码层面是如何实现和运行的 以此为基础，研究一下如何添加ADC评估算法，诸如在哪层页面如何添加，构建初步的实现方案

### 项目架构探索

#### AlgorithmManage 算子管理

* 算子相关操作
* 获取算子信息
* 验证算子
* 生成计算任务
* 生成并管理python脚本
* 管理xml文件

#### AnalysisResult 分析结果

* 分析过程相关操作
* 添加展示窗口，图形窗口并进行删除修改操作
* 添加数据
* 添加指标信息
* 截图

#### DataAnalysis 数据分析

* 数据展示窗口的相关操作

#### DBService 数据库服务

* 数据库相关操作
* 开启和关闭数据库
* 执行输入的sql语句
* 返回相应的数据库相关参数

#### EvaluationProcess 评估过程

* 依据工程id唤起其他操作
* 依据id获取工程数据
* 获取数据及数据集信息

#### EvaluationProject 评估主项目

#### EvaluationResult 评估结果

* 数据评估结果展示
* 初始化结果展示窗口
* 展示计算结果

### 项目添加ADC算法的主要思路

#### 算法所需参数

* **A 可用度**
* 一个需要用户输入的向量，长度为n(状态数量)
* 参数1 任务开始时可能出现的各种状态(正常状态，故障状态)
* 参数2 任务开始时出现各种状态的概率
* 或者
* 参数1 系统处于正常状态的时间
* 参数2 系统处于故障状态的时间
* 参数3 系统进行故障维修的时间
* 注1: 依据不同的参数传入，进行不同的可用度计算
* 注2: 状态使用故障程度从低到高排列，使后续生成一个更便于查看的可靠度矩阵
* **D 可靠度**
* 一个需要用户输入的矩阵，大小为n\*n
* 参数1 任务进行过程中，系统可能在状态之间相互转换，记录两两转换的概率
* 注1: 矩阵中任意一行的元素和为1
* 注2: 若安装角标值越大故障越严重的方式排列，并且状态转为故障后不能自行修复，则矩阵变成上三角矩阵
* **C 效能值**
* 取决于系统本身的性能和具体作战情况 创建方式可以通过层次分析法、专家评估法来确定

#### 算法计算方式

##### 可用度计算

依据概率判断系统处于正常的概率和处于故障的概率 (在系统出现两次故障之间正常运作的时间和系统从故障转到正常的时间计算) T1为正常开始使用到故障开始时间 T2为故障开始时间到维修开始时间 T3为维修开始时间到维修结束时间

* MTTF (Mean Time To Failure，平均无故障时间)，指系统无故障运行的平均时间，取所有从系统开始正常运行到发生故障之间的时间段的平均值。

**MTTF =∑T1/ N**

* MTTR (Mean Time To Repair，平均修复时间)，指系统从发生故障到维修结束之间的时间段的平均值。

**MTTR =∑(T2+T3)/ N**

* MTBF (Mean Time Between Failure，平均失效间隔)，指系统两次故障发生时间之间的时间段的平均值。

**MTBF =∑(T2+T3+T1)/ N**

很明显:**MTBF= MTTF+ MTTR**

系统正常概率 **Pzc = MTBF / (MTBF + MTTR)**

系统故障概率 **Pgz = MTTR / (MTBF + MTTR)**

系统的可用度可以用 **(∑T1 / N) / (∑(T2+T3+T1)/ N) = (∑T1) / ∑(T2+T3+T1)** 表示

##### 可靠度计算

执行任务保持正常的概率Pnor服从指数分布

指数分布：**exp(-γT) = 1/(-γT) \* e ^ ( x / (γT) )**

**Pnor=exp(-γT)**

T为任务时间，γ为故障率 **γ=1/MTBF**

> **MTBF = ∑(T2+T3+T1)/ N**

系统转为某种故障的概率 = 系统不同部件处于不同状态的概率乘积

> 比如系统左引擎故障概率为γ1, 右引擎正常概率为γ2, 系统处于左引擎正常右引擎故障概率为γ1\*γ2

### 个人算法研究方向

* 一致性修正算法：粒子群优化算法
* 归一值的计算方式
* 程序中python算子是如何运作的
* 程序数据分析中，映射规则和分析规则是如何使用的

### 后续功能修改大致方向

* 在设定权重值时，若权重之和要求为一，可以依据之前已经填入的数据，自动平均设置剩下的权重值
* ~~删除评估方案时，无法删除(且不提示删除错误的原因)~~
* 数据预处理中，当未添加数据源时，显示无数据源，但是该数据源也可作为一个数据源打开使用，并且在创建数据源后依旧存在